ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 105-110

Minimerror: A Perceptron Learning Rule that Finds the Optimal Weights

Mirta B. Gordon and Dominique Berchier

SPSMS/DRFMC - Centre d'Etudes Nucléaires de Grenoble
85X - 38041 Grenoble Cédex - France

Abstract. We present simulation results of a new learning rule for binary perceptrons that finds the optimal weights, whose properties were predicted by Gardner and Derrida [1]. The algorithm proceeds by gradient descent on a cost function that measures the mean number of errors on the learning set, at a given temperature. Simulation results are compared to the theoretical predictions.

1.Introduction

The perceptron is the simplest architecture for a neural network. It consists of N input units labelled i (i=1,2,...,N), all connected to an output unit via synaptic weights of strenghts J_i. We are concerned here with binary perceptrons, that is, whose units can be either in state σ_i=1 or σ_i=-1. If the weights J_i take suitable values, a perceptron can perform linear separations of the input patterns [2]. Although this is not a very high performance, perceptrons are important because more involved architectures can be constructed taking them as building blocks. For example, one of the most promising approaches to multilayered neural networks are constructivistic algorithms that add successively new perceptrons to the network in hidden layers until the outputs to all the patterns of the learning set are correct [3], [4], [5].

Given weights defined by the weight vector $\mathbf{J}=\{J_i, 1\le i\le N\}$, and given a learning set of P patterns, defined by their inputs $\xi^\mu=\{\xi_i^\mu; 1\le\mu\le P; 1\le i\le N\}$ and the corresponding outputs $\tau^\mu; 1\le\mu\le P$; the stability of a pattern μ is:

$$\gamma^\mu = \tau^\mu \sum_{i=1}^{N} J_i \xi_i^\mu \qquad (1.1)$$

Patterns are well learned if their stabilities are positive. Optimal weights are those that maximise the stabilities of the patterns of the learning set, under the constraint that their norm remains constant. This restriction ensures that high stabilities are not produced by simply scaling the synaptic weights. Usually the norm is fixed to $\sqrt{N}$:

$$|\mathbf{J}|^2 = \sum_{i=1}^{N} J_i^2 = N \qquad (1.2)$$

To have a geometrical picture of what the stabilities are, consider the hypercube of side length 2 centered on the origin, in dimension N. Each apex is one possible input pattern, and has a sign equal to the corresponding output. Then, any weight vector **J** defines a hyperplane normal to **J** passing through the origin, assigning a positive sign to the half space into which **J** points. The stability of a pattern is the distance from the

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 105-110

corresponding apex to the hyperplane. It is positive if the corner is in the half space corresponding to its sign, negative otherwise. Therefore, if weights are optimal, the corresponding hyperplane should be as far as possible from its nearest corners (positive and negative) belonging to the learning set. The properties of binary perceptrons with optimal weights have been analysed with methods of Statistical Mechanics [6],[1] applied to the phase space of the synaptic weights, under the normalisation constraint. The theoretical results predict the capacity $\alpha=P/N$ of an optimal perceptron, that is, the maximum number of patterns chosen at random that the perceptron is able to classify correctly, in the limit of a very large number of input units. Beyond the well known result $\alpha=2$ [7], within the statistical mechanical approach it was possible to determine the capacity when the stabilities are forced to be larger than an imposed value κ [6], the distribution of stabilities[8], the minimum number of errors done by the perceptron beyond its limit of capacity [1], etc. However, this approach does not give any recipe of how to determine these optimal weights.

The weights of a perceptron are usually determined with a learning rule, an algorithm that gives the values of the J_i that hopefully make the stabilities (1.1) positive, or higher than the imposed stability κ, for all the patterns of the learning set. If $\kappa=0$, the Perceptron learning rule [2] is able to find a solution to this problem if a solution exists. Higher performance rules exist in this case, even for $\kappa>0$, and in particular Minover [9] gives the optimal weights, which maximise the stabilities (1.1). However, these rules are unable to detect if there is a solution, and may get stucked in a loop if a solution does not exist. Other learning rules exist that can handle with the learning problem even if there is no set of weights making no errors on the learning set. These rules either detect the absence of a solution, and stop [10], [11], or stop giving a 'reasonable' set of weights [12], [13],.but they cannot assure that the set of weights found minimise the number of errors.

In this paper we present a new learning rule, Minimerror, that finds the optimal weights with a gradient descent search. If a solution exists, the rule gives the weights that maximise the stabilities of the learning set. If the problem has no solution, our algorithm gives the weights that minimise the number of errors on the learning set. The cost function that has to be minimised may be deduced from statistical mechanics [14] and is the mean number of errors of the perceptron on the learning set, at a given temperature T. The optimal perceptron properties have been calculated precisely by counting the number of errors at T=0, which is a stepped function of the J_i. By the introduction of a temperature,we can calculate the mean number of errors, which is a smooth continuous function of the weights, making a gradient descent search possible. During the gradient descent, this temperature is slowly decreased, like in a simulated annealing, eventually reaching T=0.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 105-110

2.The number of errors at finite temperature

Statistical mechanics may be used as a tool for generating learning rules for perceptrons [14] and for neural networks of more envolved architectures [15]. Consider the space of synaptic weights: each point **J** in this space corresponds to one possible perceptron, having stabilities (1.1) for the learning set. We associate to each point **J** a set of P energy levels, the energy of each level being the stability γ^μ of the corresponding pattern. In order to count the number of errors done by perceptron **J**, we suppose that we have a reservoir of particles. If the energy of a level is lower than the stability κ we want to impose to the network, then we put one particle on that level. Clearly, the number of errors is just the number of particles. The same reasoning can be made at temperature T by occupying each level with a probability given by the Boltzmann factor: $p = Z^{-1}\exp\left[\left(\gamma^\mu - \kappa\right)/T\right]$, where Z is the partition function. In this case it is easy to show [14] that the total mean number of errors at temperature T is:

$$\langle n\rangle = \sum_{\mu=1}^{P}\left\{1+\exp\left[\left(\gamma^\mu - \kappa\right)/T\right]\right\}^{-1} \qquad (2.1)$$

Therefore, the weights that minimise (2.1) can be determined by gradient descent, starting from any initial state **J**:

$$J_i \leftarrow J_i + \delta J_i$$
$$\delta J_i = -\varepsilon\frac{\partial\langle n\rangle}{\partial J_i} \qquad (2.2)$$

Introducing (2.1) into (2.2) gives the learning rule:

$$\delta J_i = -\frac{\varepsilon}{4T}\sum_{\mu=1}^{P}\left\{\cosh\left[\left(\gamma^\mu - \kappa\right)/2T\right]\right\}^{-2} \qquad (2.3)$$

Each pattern is learnt with an intensity, or weighting factor, that depends on its stability. Patterns with stabilities within an interval of width 2T close to κ have strong weighting factors, whereas patterns whose stabilities are far apart almost do not contribute to modify the synaptic weights. Beginning the learning session at high temperature allows to learn all the patterns with almost the same intensity, like with Hebb's rule, and cooling down slowly during the learning session increases the importance of those patterns that are close to the hyperplane, which are then selectively learned. This seems a clever strategy: not only patterns with stability slightly less than κ are learned, but also those with stability slightly higher than κ, to eventually avoid their destabilisation. Patterns with stabilities far apart from the hyperplane will not be greatly affected by the small incremental changes (2.3), and the rule give them low weighting factors on learning.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 105-110

3.Implementation and simulation results

In order to compare with the theoretical predictions, in all the simulations the learning sets are composed of random patterns. Preliminary tests of algorithm (2.3) showed that very frequently it got stacked in local minimæ during the annealing procedure. This happens because there are local minimæ of (2.1), which correspond to hyperplanes just at the barycenter of the patterns within a distance 2T from them. Inspired by works on spin systems [16], we introduced wo temperatures: T_- for negative stability patterns, and T_+ for positive stability patterns, with $T_->T_+$. That is, non learned patterns are considered at higher temperature than stable ones. In the simulations, the ratio T_-/T_+ was kept constant, a value of $T_-/T_+=6$ was found to give good results. The synaptic weights were modified according to (2.3) until the number of errors was constant, and then T+ was decreased following:$\delta(T_+)^{-1} \equiv 10^{-3}$ until the number of errors became again constant. We replaced the prefactor $\varepsilon/4T$ by a single parameter Δ, which was initialized at $\Delta\cong 0.2$ each time T_+ was decreased, and then was slowly decreased during the search at constant temperature.

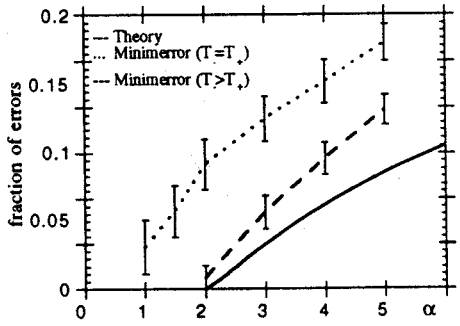

Fig. 1 Fraction of errors versus α: simulation results averaged over 30 samples.

Figure 1 shows the fraction of errors $f = \langle n \rangle / P$ as a function of α, for $\kappa=0$, obtained on a perceptron of N=50 input units, together with the theoretical prediction [1]. The difference between theory and simulations may be due to size effects. Moreover, the difference decreases for larger N, as is shown on Fig.2 for $\alpha=4$ as a function of N.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 105-110

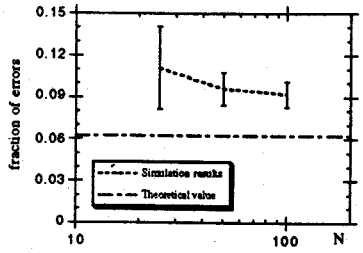

Fig.2 Fraction of errors versus N for α=4.

The distribution of stabilities for α=0.5, 2 and 4 are shown on Figure 3. Agreement with the theoretical predictions is very good [8]. In our numerical simulations, rounding of the delta functions predicted by the theory is apparent, because of the finite size of the samples.

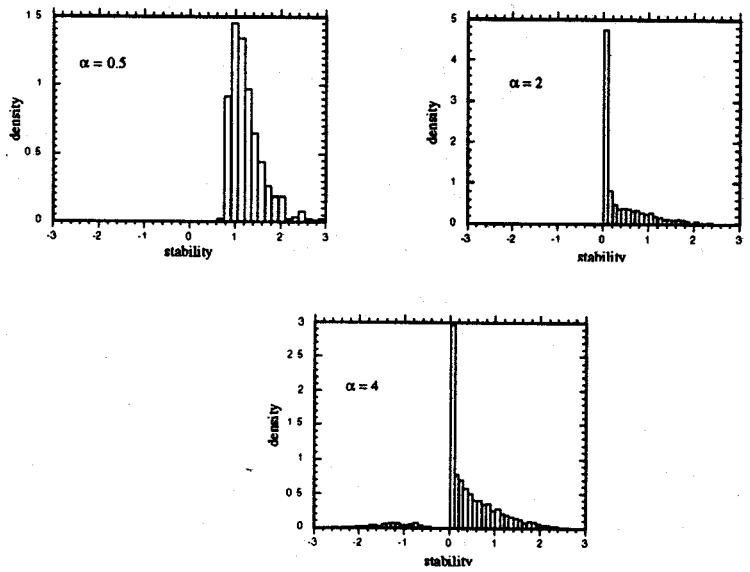

Fig.3 Stabilities distributions for different α, N=30, averages over 30 samples.

4. Conclusion

We presented simulation results of a new learning algorithm for perceptrons, Minimerror, that finds the optimal weights even when the learning set is not linearly separable. Comparisons were made with theoretical predictions on learning random patterns. We are currently testing the performance of the algorithm on generalisation tasks. Minimerror is useful to generate small feedforward networks by constructivistic algorithms [5], and results of its performance in this case will be published elsewhere.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 105-110

References

1. E. Gardner and B. Derrida: Optimal storage properties of neural network models. J.Phys.**A**, 21, 271 (1988)
2. M. Minsky and S. Papert, 'Perceptrons', (MIT Press, Cambridge, 1988)
3. M. Mezard and J.-P. Nadal: Learning in feedforward layered neural networks: the tiling algorithm. J.Phys.**A**, **22**, 2191-2203 (1989)
4. D. Martinez and D. Esteve: The Offset Algorithm: Building and Learning Method for Multilayer Neural Networks. Europhys. Lett., **18**, 95-100 (1992)
5. P. Peretto and M. B. Gordon, 'Monoplane: a constructive learning algorithm for one-hidden layer feedforward neural networks', Neural Networks for Computing (1992)
6. E. Gardner: Maximum storage capacity in neural networks. Europhys. Lett., **4**, 481-485 (1987)
7. T. M. Cover: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans. on Electronic Computers, **EC14**, 326-334 (1965)
8. D. J. Amit, M. R. Evans, H. Horner and K. Y. M. Wong: Retrieval phase diagrams for attractor neural networks with optimal interactions. J. Phys. **A**: Math.Gen., **23**, 3361-3381 (1990)
9. W. Krauth and M. Mezard: Learning algorithms with optimal stability in neural networks. J.Phys.**A**, **20**, L745-L752 (1987)
10. D. Nabutovsky and E. Domany: Learning the Unlearnable. Neural Computation, **3**, 604-616 (1991)
11. P. Rujan: A fast method for calculating the perceptron with maximal stability. Preprint, (1992)
12. S. I. Gallant: Perceptron-based Learning Algorithms. IEEE Transactions on Neural Networks, **1**, 179-191 (1990)
13. M. Frean: A "thermal" perceptron learning rule. Neural Computation, 4, (1992)
14. M. B. Gordon, P. Peretto and D. Berchier: Learning Algorithms for Perceptrons from Statistical Physics. J.Physique, (1993)
15. M. B. Gordon, P. Peretto and M. Rodriguez-Girones: Learning in feed-forward neural networks by improving the performance. Physica **A**, 185, 402-410 (1992)
16. P. Gawiec and D. Grempel: Numerical study of the ground-state properties of a frustrated XY model. Phys.Rev.B, 44, 2613-2623 (1991)